安定化理論に関するいくつかの注意 (Computer Algebra : Algorithms, Implementations and Applications)

AUTHOR(S):

白柳, 潔

CITATION:

白柳, 潔. 安定化理論に関するいくつかの注意 (Computer Algebra : Algorithms, Implementations and Applications). 数理解析研究所講究録 2002, 1295: 189-196

ISSUE DATE:

2002-11

URL:

http://hdl.handle.net/2433/42618

数理解析研究所講究録 1295 巻 2002 年 189-196

安定化理論に関するいくつかの注意

NTT コミュニケーション科学基礎研究所 白柳 潔 (Kiyoshi Shirayanagi)

1 はじめに

1988 年、近似代数が佐々木氏 [12] によって提唱されて以来、数値数式融合計算なる分野 [9, 8] が盛んである。欧米でも SNAP (Symbolic-Numeric Algebra for Polynomials) の研究がますます活発になってきている [7, 24]。一方、この潮流の中で筆者と M. Sweedler が 1995 年に発表したアルゴリズムの安定化理論 [21] も、年々、さまざまな計算に応用されつつある。近似代数や SNAP が、「ノイズのある入力から、意味のある出力を探し求める」という立場であるのに対し、安定化理論では、「正確な入力から、近似計算によって、正確な出力に接近する」という立場をとっている。安定化理論の手法で安定化されたアルゴリズムを使うと、十分高い精度桁で近似計算を行なえば、元のアルゴリズムを正確演算で実行したときの過程を完全にシミュレートできる。換言すれば、近似入力が真の入力に収束すれば、対応する近似出力も真の出力に収束するようになる。

本論は、安定化理論を簡単に復習した後、本理論のこれまでの応用例や得られた知見などを基に、いくつかの注意を記し、今後のさらなる研究推進に資することを目的とする。

2 安定化理論の復習

次のアルゴリズムを対象に安定化理論の復習を簡単に行う。詳細は [17] を参照されたい。

- データは、すべて多項式環 $R[x_1, \ldots, x_m]$ の元からなる。R は実数体の部分体である。
- データ間の演算は、$R[x_1, \ldots, x_m]$ 内の加減乗または剰余計算である。
- データ上の述語は、不連続点をもつとすればそれは 0 のみである。

述語の不連続点が 0 という意味は、If "$C = 0$" then ... else ... のように、値が 0 か否かによって分岐が別れることである。上記クラスのアルゴリズムを、不連続点 0 の代数的アルゴリズムとよぶ。ほとんどの数式処理のアルゴリズムはこのクラスに入るか、このクラスのアルゴリズムに変換可能である。

さて、安定化の 3 つのポイントは、

- アルゴリズムの構造は変えない。

- データ領域において、ふつうの係数を区間係数に変える。
- 述語の評価の直前で、区間係数のゼロ書換えを行なう。

である。すなわち、安定化されたアルゴリズムは次のようになる。

区間領域 データ領域は区間係数多項式の集合。区間係数は $[A,\alpha]$ なる形で、$A \in R$, α は非負の実数。$[A,\alpha]$ は集合 $\{x \in R \mid |x-A| \leq \alpha\}$ を意味する。

区間演算 二項演算 $* \in \{+,-,\times,/\}$ に対し、

$$[A,\alpha] * [B,\beta] = [A * B, \gamma_*].$$

ここに、γ_* は次を満たす。

$$|x-A| \leq \alpha, |y-B| \leq \beta \Rightarrow |x*y - A*B| \leq \gamma_*.$$

ゼロ書換え 不連続点0をもつ述語を評価する直前で、各区間係数 $[C,\gamma]$ に対し、

$|C| \leq \gamma$ ならば $[C,\gamma]$ を [0,0] に書き換えよ。

$|C| > \gamma$ ならばそのまま。

今、入力 $f \in R[x_1,\ldots,x_m]$ を

$$f = \sum_{i_1,\ldots,i_m} a_{i_1\ldots i_m} x_1^{i_1} \cdots x_m^{i_m}$$

と表したとき、f に対する近似列 $\{Int(f)_j\}_j$ を

$$Int(f)_j = \sum_{i_1,\ldots,i_m} [(a_{i_1\ldots i_m})_j, (\alpha_{i_1\ldots i_m})_j]\, x_1^{i_1} \cdots x_m^{i_m}$$

で定義する。ここに、すべての $i_1,\ldots,i_m$ について、

$$|a_{i_1\ldots i_m} - (a_{i_1\ldots i_m})_j| \leq (\alpha_{i_1\ldots i_m})_j \text{ for } \forall j$$

$$(\alpha_{i_1\ldots i_m})_j \to 0 \text{ as } j \to \infty$$

このとき、単に

$$Int(f)_j \to f$$

と書く。

さて、$\mathcal{A}$ を安定化したアルゴリズムを $Stab(\mathcal{A})$ と書くと、次が安定化理論の基本定理である。

定理 1 (係数収束) $\mathcal{A}$ は不連続点 0 の代数的アルゴリズムで、入力 $f \in R[x_1,\ldots,x_m]$ に対し正常終了するとせよ。このとき、f に対する任意の近似列 $\{Int(f)_j\}_j$ に対し、ある n が存在して、$j \geq n$ ならば、$Stab(\mathcal{A})$ は $Int(f)_j$ に対し正常終了し、

$$Stab(\mathcal{A})(Int(f)_j) \to \mathcal{A}(f).$$

次章から、安定化理論に関して、順不同にいくつかの注意を記述する。

3 陽に条件文がない場合

2章で、安定化の3つ目の基本は、条件文において「区間係数の書換え」を行なうことであると書いた。すなわち、例えば、アルゴリズムの中に係数Cに対して、

If $C=0$ Then ...

という命令があったら、安定化したアルゴリズムにおいては、Cに対応する区間係数$[C_i,\gamma_i]$に対して、述語の評価（ゼロの判定）をする直前に、「ゼロ書換え」を行う。つまり、

$$|C_i|\leq\gamma_i \text{ ならば } [C_i,\gamma_i] \text{ を } [0,0] \text{ に書き換えよ。}$$

しかし、一般にアルゴリズムの中には、陽にif文が現れていなくても陰に述語が隠されている場合がある。例えば、一変数多項式の主項をとる操作

$$leadmon:\sum a_ix^i\mapsto x^{\max\{i|a_i\neq 0\}}$$

は、係数がゼロでない項のうち最大次数の項をとる操作であり、ゼロか否かを問う述語が隠されている。このような場合に備えて、「各多項式演算の実行の**直後**にゼロ書換え」するのが、一つの便法である。上記の例で言えば、区間係数多項式に対して$leadmon$は、

$$leadmon:\sum[a_i,\alpha_i]x^i\mapsto x^{\max\{i|[a_i,\alpha_i]\neq 0\}}$$

となるが、ある多項式演算の結果、$\sum[c_i,\gamma_i]x^i$が得られ、それの$leadmon$をとるときは、各区間係数$[c_i,\gamma_i]$にゼロ書換えを行ってから$leadmon$にかける。これによって、陰に入っている述語の前にゼロ書換えが行なわれるようになるのである。

4 台収束の重要性

多くの場合、係数収束より強い「台収束」が重要となる。まず「台」を定義する。

$$f=\sum_{i_1,\ldots,i_m}a_{i_1\ldots i_m}x_1^{i_1}\cdots x_m^{i_m}$$

の台とは、集合

$$\{x_1^{i_1}\cdots x_m^{i_m}\mid a_{i_1\ldots i_m}\neq 0\}$$

で、これを$Supp(f)$と記す。

$f_j\to f$が台収束するとは、係数収束し、かつ、ある**有限**のjで、$Supp(f_j)=Supp(f)$となることである。

（例）$j\to\infty$のとき、$\frac{j-1}{j}\cdot x^2+x+1\to x^2+x+1$は台収束である。$\frac{1}{j}\cdot x^2+x+1\to x+1$は係数収束であるが台収束ではない。

ここで、次の方法によって台収束が実現できる。

$Stab(\mathcal{A})$の実行後に、各区間係数にゼロ書換えを施す。

このアルゴリズムを$Stab(\mathcal{A})_R$と書く。このとき、次の定理が成立する。

定理 2 (台収束) $\mathcal{A}$は不連続点0の代数的アルゴリズムで、入力$f \in R[x_1, \ldots, x_m]$に対し正常終了するとせよ。このとき、$f$に対する任意の近似列$\{Int(f)_j\}_j$に対し、ある$n$が存在して、$j \geq n$ならば、$Stab(\mathcal{A})_R$は$Int(f)_j$に対して正常終了し、$Stab(\mathcal{A})_R(Int(f)_j)$は$\mathcal{A}(f)$に台収束する。

この台収束という良い性質はもっと活用されるべきだと思う。例えば、安定化理論を使った浮動小数グレブナ基底計算によって台の候補が決まったら、未定係数法などで厳密係数のグレブナ基底を求めることも一例である。

5 これまでの応用例と知見

安定化理論がこれまでに応用された例を表に示す。区間および区間演算はすべて浮動小数点近似によるものである。各々の詳細については、表中の参考文献を参照されたい。

アルゴリズム	出力	実験者と実験年
Buchberger	グレブナ基底	(白柳 93) [15], (尾崎 94) [11], (日吉 97) [1]
Sturm	実根の個数	(関川 95) [20]
Graham	2次元および3次元 凸包	(関川 95&98) [13]
単因子法	ジョルダン標準形	(新妻 96) [6, 19]
Greville	一般逆行列	(水口 96-98) [2, 3], (村上 01) [5]
Gauss	三角行列	(村上 01) [4]
Wu	Characteristic Set	(野竹 00) [10], (白石 01) [14]

表で示した種々の実験の結果から、多くの場合共通に成り立つ知見は、以下の通りである。

1. 従来の浮動小数点計算では、高速であるが、精度桁をいくら上げても信頼できる答が得られない。
2. 正確計算では、特に係数が無理数の場合は、膨大な時間を要するか、メモリ容量を大幅に超えて計算不能に陥る。

3. 安定化手法によれば、区間解析法と同程度の速さで、普通の浮動小数点計算や区間解析法よりも信頼できる答が得られる。特に、入力が "ill-conditioned" のときに効果的である。

[18] でも注意したが、もちろん、これらの知見は同じアルゴリズムでも入力に依存し、常に安定化理論が優勢というわけではない。実際、グレブナ基底計算の場合など、入力の多項式の次数や項数があまり大きいと、好結果が得られないことがある。なぜならば、この場合は、アルゴリズムの実行中に、係数膨張というよりもむしろ多項式自体の大きさ (次数や項数) の膨張をもたらすので、浮動小数点計算あるいは区間計算といえども、それを撃退するのが難しいからである。裏を返せば、正確演算で計算したときに「係数膨張のみが計算の遅い主要な要因」と思われるとき、安定化手法は役に立つということができる。

6 よく受ける誤解

これまでに安定化理論に対して受けた誤解はいくつかあるが、代表的なものは次の2つである。

1. 安定化手法と Newton 法などの反復法とでは、解を収束させる仕組みは同じではないか？

2. ゼロ書換えは $|E| \leq \epsilon$ のとき、$[E, \epsilon]$ を $[0, 0]$ に変える。この ϵ は予め与えられる閾値か？

答えはいずれも No である。最初の誤解は、安定化手法が、出力が安定化されるまでアルゴリズムの実行を繰り返すことから来ていると思われる。しかし、反復法は得られた出力を再利用する。すなわち、それをアルゴリズムに入力してを再実行し、満足のいくまでこれを繰り返す。これに対して、安定化手法は得られた出力を再利用しない。ある出力が満足のいくものでなければ、「入力の精度」を高めてアルゴリズムを再実行する。

2番目の誤解について答えるならば、予め与えておくのは入力の区間の誤差項のみであり、それ以降の各区間の誤差項は、区間演算によって自動的に決まる。その誤差項に基づいてゼロ書換えを実行するのである。

7 精度問題の難しさ

安定化定理により、十分大きな精度 (j) になれば、収束性が保証されている。しかし、具体的にどの程度の精度で十分であるかを予め、見積もることができるだろうか？ これを精度問題と呼ぼう。実は精度問題は一般に、アルゴリズムだけではなく、入力にも依存する極めて難しい問題である。以下では、簡単な例によって、これがいかに難しい問題であるかを示そう。

$\mathcal{A}$を次のアルゴリズムとする。

$\mathcal{A}$:

入力	A, B（正の実数）
If	$A - B = 0$ then 1 else 0.

このとき、$\mathcal{A}$を安定化すると次のようになる。

$Stab(\mathcal{A})$:

入力	$[a, \alpha], [b, \beta]$
If	$\lvert a - b \rvert \leq \alpha + \beta$ then 1 else 0.

さて、$A = 0.11, B = 0.10$のとき、$\mathcal{A}(A, B) = 0$である。

$$Stab(\mathcal{A})([a_\mu, \alpha_\mu], [b_\mu, \beta_\mu])$$

はどうなるであろうか？

$\mu = 1$のとき

$[a_1, \alpha_1] = [0.1, 5 \times 10^{-2}]$　$[b_1, \beta_1] = [0.1, 5 \times 10^{-2}]$

$a_1 - b_1 = 0$　$\alpha_1 + \beta_1 = 10^{-1}$

よって$a_1 - b_1 < \alpha_1 + \beta_1$となり、$Stab(\mathcal{A})([a_1, \alpha_1], [b_1, \beta_1]) = 1$（誤答）となる。

$\mu \geq 2$のとき

$[a_\mu, \alpha_\mu] = [0.11, 5 \times 10^{-(\mu+1)}]$　$[b_\mu, \beta_\mu] = [0.10, 5 \times 10^{-(\mu+1)}]$

$a_\mu - b_\mu = 0.01$　$\alpha_\mu + \beta_\mu = 10^{-\mu}$

そこで、$\mu = 2$のときは

$a_2 - b_2 = \alpha_2 + \beta_2$なので、$Stab(\mathcal{A})([a_2, \alpha_2], [b_2, \beta_2]) = 1$でまだ正解にならない。

$\mu \geq 3$のとき

$a_\mu - b_\mu > \alpha_\mu + \beta_\mu$となって、$Stab(\mathcal{A})([a_\mu, \alpha_\mu], [b_\mu, \beta_\mu]) = 0$（正答）を得る。

入力を変えて$A_n = 0.1 + 10^{-n}, B = 0.1$のとき、同じ推論によって

$$Stab(\mathcal{A})([a_{n,\mu}, \alpha_{n,\mu}], [b_\mu, \beta_\mu])$$

は、$\mu \geq n + 1$のとき$\mathcal{A}(A_n, B)$と同じ答えを出すことがわかる。

一般に$0 < A < 1$, $0 < B < 1$, $A \neq B$のとき、最悪の場合、$Stab(\mathcal{A})$は精度桁$\mu = \lfloor |\log_{10} |A - B|| \rfloor + 1$を要する。予め$A - B$の値がわかっていれば精度桁は見積もれるが、それがわかっていればそもそも$\mathcal{A}$の安定化を考える必要はないことになる。

8　おわりに

安定化理論についていくつかの注意を行った。その他、安定化によって得られた結果が正しいかどうかを検証する研究（例えば計算履歴法[16, 22]の研究）が重要であること、また、[18, 23]でも新たな使い方を示したように、安定化理論にはさまざまな応用の可能性があることを改めて強調しておきたい。

参 考 文 献

[1] 日吉: 計算代数・計算幾何における近似計算の利用法, 東京大学 (計数工学)1996 年度修士論文 (1997).

[2] Minakuchi, H., Kai, H., Shirayanagi, K. and Noda, M-T.: Algorithm Stabilization Techniques and their Application to Symbolic Computation of Generalized Inverses, *Electronic Proc. of IMACS-ACA '97* (1997). (http://www.math.unm.edu/ACA/1997/Proceedings/MainPage.html)

[3] 水口, 白柳: 安定化理論の一般逆行列への応用, 数式処理, **6** 1 (1997), 45–46.

[4] 村上, 甲斐, 野田 : 区間演算によるハイブリッド有理関数近似と安定化理論について、京大数理研講究録 (本号) に掲載予定.

[5] 村上, 白柳 : 安定化理論の文字認識への応用、数式処理, **8** 4 (2002), 11–13.

[6] 新妻, 白柳: 浮動小数ジョルダン標準形の安定化, 数理解析研究所講究録, **986** (1997), 195–205.

[7] 野田: SNAP96 (Workshop on Symbolic-Numeric Algebra for Polynomials), 数式処理, **5** 1 (1996), 58–60.

[8] 野田, 甲斐 : 数式処理と数値計算 – いかに結合させるか？, 情報処理 **39** 2 (1998), 105–110.

[9] Noda, M-T. and Sasaki, T.: Approximate GCD and Its Application to Ill-conditioned Algebraic Equations, *J. Computational and Applied Mathematics*, **38** (1991), 335–351.

[10] 野竹, 甲斐, 支, 野田 : Wu's method の浮動小数化, 京大数理研講究録 **1199** (2001), 1–9.

[11] 尾崎, 白柳: Maple の Interval Package を用いた浮動小数グレブナ基底の計算, 数理解析研究所講究録, **920** (1995), 38–52.

[12] 佐々木: 近似的代数計算法, 数理解析研究所講究録, **676** (1988), 307–319.

[13] 関川, 白柳: 凸包アルゴリズムの安定化, 日本数式処理学会第 4 回大会配布資料 (1995).

[14] 白石 : 安定化した Wu's method のロボット制御への応用, 京大数理研講究録 (本号) に掲載予定.

[15] Shirayanagi, K.: Floating Point Gröbner Bases, *Mathematics and Computers in Simulation*, **42** 4-6 (1996), 509–528.

[16] 白柳: 安定化理論と出力の妥当性について, 数式処理, **5** 1 (1996), 46–47.

[17] 白柳: アルゴリズムの安定化理論, 数式処理, **5** 2 (1997), 2–21.

[18] 白柳: 安定化理論の新しい使い道, 数理解析研究所講究録, **1038** (1998), 170–176.

[19] 白柳, 新妻: 実多項式行列スミス標準形の安定な浮動小数点計算法, 情報処理学会論文誌, **40** 3 (1999), 1006–1017.

[20] 白柳, 関川: ゼロ書換えに基づいた区間法と Sturm のアルゴリズムへの応用, 電子情報通信学会論文誌 A, **J80-A** 5 (1997), 791–802.

[21] Shirayanagi, K. and Sweedler, M.: A Theory of Stabilizing Algebraic Algorithms, *Technical Report 95-28, Mathematical Sciences Institute, Cornell University* (1995), 92 pages.

[22] Shirayanagi, K. and Sweedler, M.: Automatic Algorithm Stabilization, *ISSAC'96 poster session abstracts* (1996), 75–78.

[23] Shirayanagi, K. and Sweedler, M.: Remarks on Automatic Algorithm Stabilization, *Journal of Symbolic Computation* **26** 6 (Dec. 1998), 761–766.

[24] Special Issue of Symbolic-Numeric Algebra for Polynomials, *Journal of Symbolic Computation* **26** 6 (Dec. 1998).